Jacob Le Nguyen
Professor Christopher Smith
CS 497

Assignment 5

Abstract

In this assignment we will be writing a shell script to provide naive encryption features, file reading, file writing, string manipulation, and a looping menu. While we will be demonstrating how the menu and encryption functions should work in this assignment, the shell script should be written by yourself based off of the example and run through. After finishing the assignment we should have a solid understanding of how shell scripts should run and work.

Introduction

We will be using linux ubuntu, command line, terminal, and bash commands.

Commands used:
#edit script file
nano assignment5.sh

#run script file
bash assignment5.sh

Summary of Results

```
jacob@jacob-VirtualBox:~$ nano assignment5.sh
```

1. We want to begin by starting up linux ubuntu. After doing so we want to login and navigate to the terminal so we can begin the assignment. When arriving at the terminal we want to enter in the command “nano assignment5.sh” in this case assignment5 being

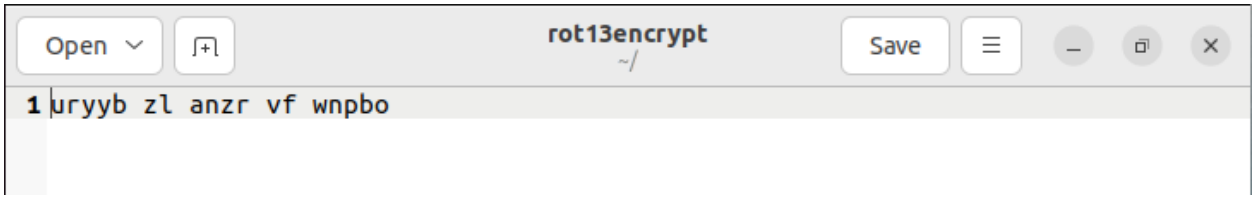
the name of the file we are creating and sh is a shell script file using the nano editor. We then want to write the shell script according to the homework pdfs specifications such as providing a looping menu, the user may choose to read a message from a file or write one, prompt user to use rot13 or cc encryption, prompt user for output file name, and execute the options.

```
jacob@jacob-VirtualBox:~$ bash assignment5.sh
```

```
jacob@jacob-VirtualBox:~$ nano assignment5.sh
jacob@jacob-VirtualBox:~$ bash assignment5.sh
Enter in an option to choose 1-3
1. Read a file
2. Type a message
3. Exit
```

2. Next to run the script file we created we want to call “bash assignment5.sh” which will run the script file we have just created. The following will be a walkthrough of the script I made, but we can use it as an example.

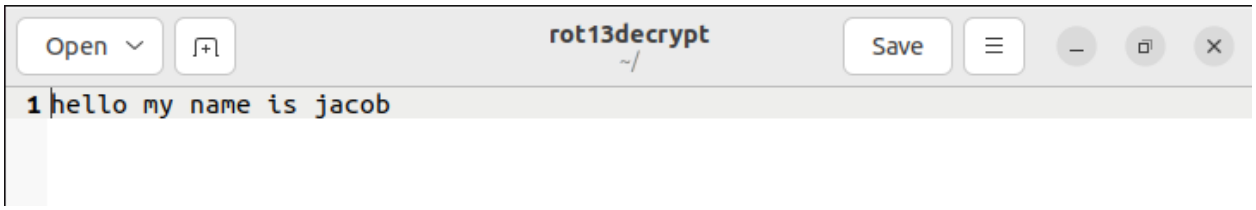
```
jacob@jacob-VirtualBox:~$ bash assignment5.sh
Enter in an option to choose 1-3
1. Read a file
2. Type a message
3. Exit
2
Please enter in some text
hello my name is jacob
Choose rot13 or cc
rot13
Encrypt or Decrypt
Encrypt
rot13 and encrypt
Please enter output file name
rot13encrypt
Enter in an option to choose 1-3
1. Read a file
2. Type a message
3. Exit
3
jacob@jacob-VirtualBox:~$
```



3. After the program is run we want to choose the second option which will allow us to enter in some text. Next we want to choose the type of encryption which in the example above is rot13. Then we want to choose to encrypt or decrypt, which I then chose to encrypt. Then we choose an output file name and then we exit the menu. We want to

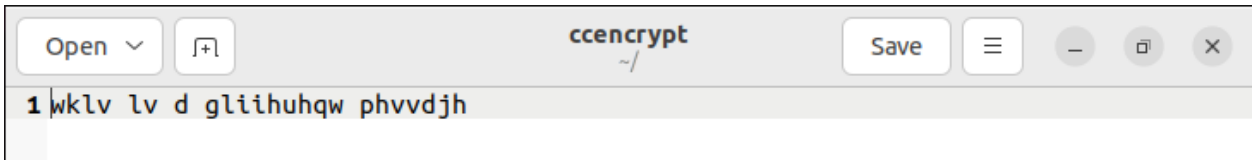
navigate through the file system until we find the “rot13encrypt” file we had just created. After displaying it we can see the contents are encrypted with the message we just entered in.

```
jacob@jacob-VirtualBox:~$ bash assignment5.sh
Enter in an option to choose 1-3
1. Read a file
2. Type a message
3. Exit
2
Please enter in some text
uryyb zl anzr vf wnpbo
Choose rot13 or cc
rot13
Encrypt or Decrypt
Decrypt
rot13 and decrypt
Please enter output file name
rot13decrypt
Enter in an option to choose 1-3
1. Read a file
2. Type a message
3. Exit
3
jacob@jacob-VirtualBox:~$
```



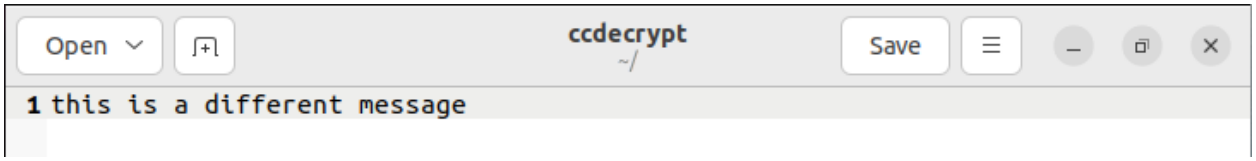
4. Then we want to demonstrate the same method but decrypting. Before exiting the encrypted file we want to copy the contents to use for this demonstration. We want to run the shell script we have created again then we again want to choose option 2. Once the enter text prompt comes up we want to then paste the encryption files results in there. Then we want to choose rot13 encryption, after doing so we want to choose decrypt. When the next prompt shows we want to name the file which in this case I did “rot13decrypt”. We then want to navigate to our files which should then contain the output file we just made, upon opening we should see the message was decrypted and stored in the file.

```
jacob@jacob-VirtualBox:~$ bash assignment5.sh
Enter in an option to choose 1-3
1. Read a file
2. Type a message
3. Exit
2
Please enter in some text
this is a different message
Choose rot13 or cc
cc
Encrypt or Decrypt
Encrypt
cc and encrypt
Please enter in number of shifts 1-25
3
Please enter output file name
ccencrypt
Enter in an option to choose 1-3
1. Read a file
2. Type a message
3. Exit
3
jacob@jacob-VirtualBox:~$
```

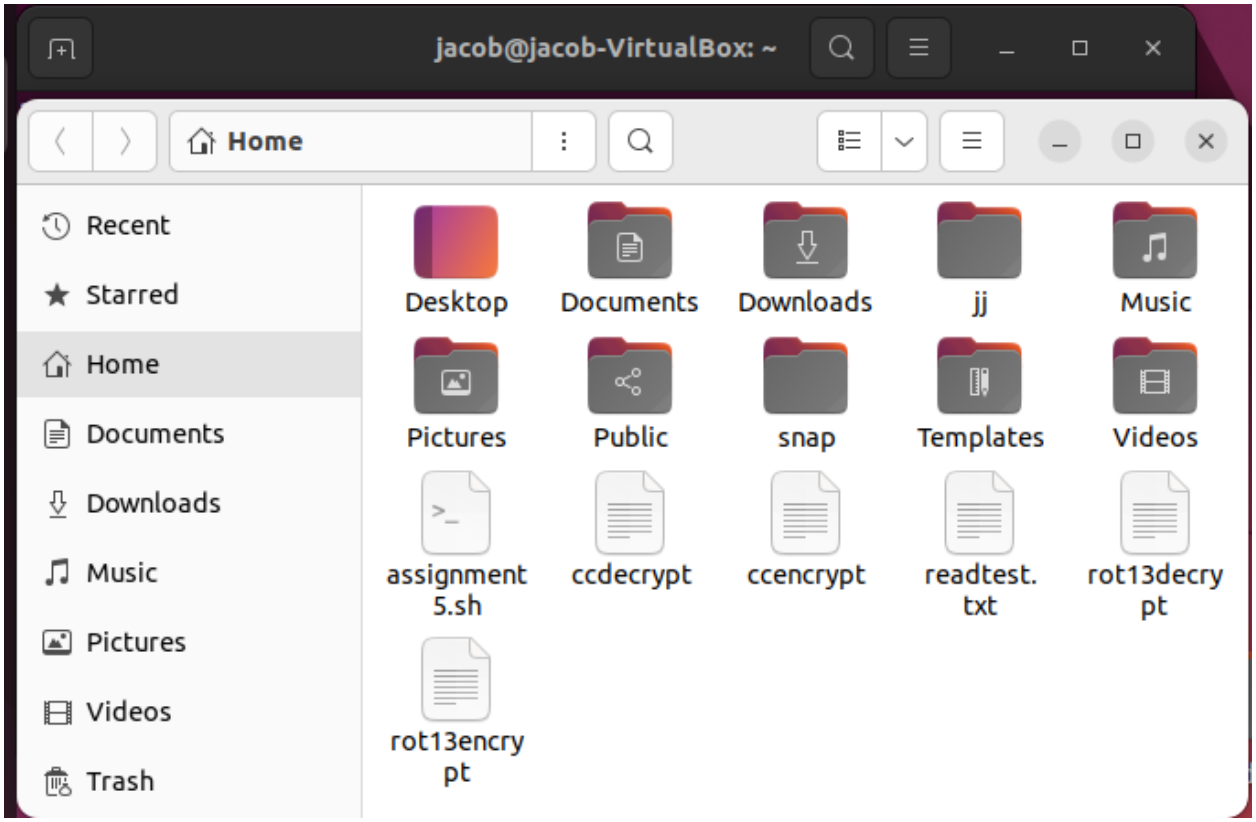


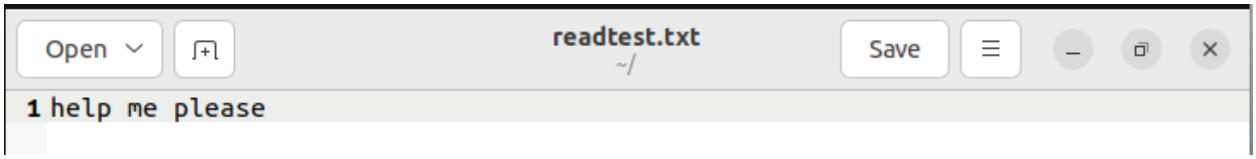
5. In this step we again want to run the script file, after doing so we choose option 2 and enter in a new message different from the last example. Once the message is entered in we then want to choose the cc encryption, then we are prompted with either encrypt or decrypt, we want to choose encrypt. After doing so we should be prompted to enter in the number of shifts which will change the encryption according to your answer. We then need to choose an output file name which I chose “ccencrypt”. Then exit the program by choosing 3 and then navigate to the encrypt file which should be in our file system. In the file we can see the message is encrypted. We then want to copy the contents and move onto the next step. Also, remember the number of shifts which are needed for decryption.

```
jacob@jacob-VirtualBox:~$ bash assignment5.sh
Enter in an option to choose 1-3
1. Read a file
2. Type a message
3. Exit
2
Please enter in some text
wklv lv d gliihuhqw phvvdjh
Choose rot13 or cc
cc
Encrypt or Decrypt
Decrypt
cc and decrypt
Please enter in number of shifts 1-25
3
Please enter output file name
ccdecrypt
Enter in an option to choose 1-3
1. Read a file
2. Type a message
3. Exit
3
jacob@jacob-VirtualBox:~$
```



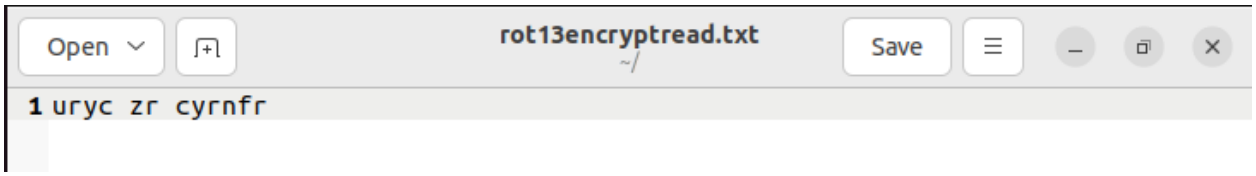
6. We want to open the terminal again and run the script. When prompted with the menu we want to choose option2. When given the option to enter text we want to paste the contents of the encryption file from the last step into the command line option. We then want to choose cc and then the option to decrypt. It should give you the option for shifts which is when we enter in the previous number of shifts we chose so that we can decrypt the message. In this case it was 3 shifts. Then we name the output file which I chose “ccdecrypt” as the name, then exit the program. Once closed we want to navigate to the ccdecrypt file, this should contain the message we have encrypted earlier.





7. Now that we have demonstrated all outcomes and processes for entering a user decided text we want to read from a file. To help demonstrate this without using the old files we want to create a new file which in this case is called “readtest.txt”. We next need to populate the file with some message for later encryption and decryption. In this case I chose “help me please”.

```
jacob@jacob-VirtualBox:~$ bash assignment5.sh
Enter in an option to choose 1-3
1. Read a file
2. Type a message
3. Exit
1
Please enter in file name
readtest.txt
Choose rot13 or cc
rot13
Encrypt or Decrypt
Encrypt
rot13 and encrypt
Please enter output file name
rot13encryptread.txt
Enter in an option to choose 1-3
1. Read a file
2. Type a message
3. Exit
3
jacob@jacob-VirtualBox:~$
```

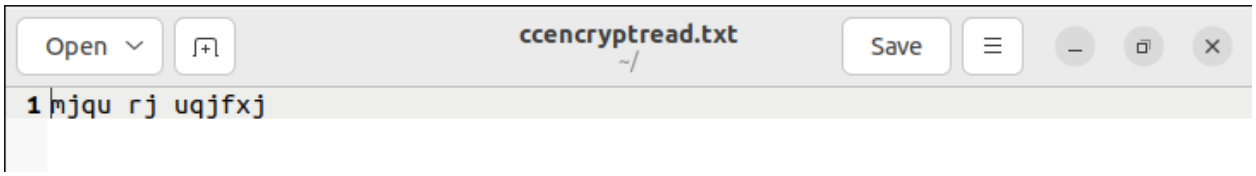


8. Next we are going to demonstrate reading a message from a file using the text file we just created from the last step. First we want to start the script file and run it. Then we choose option 1 which is read from a file. Then we want to enter in the name of the file we just created in the last step. Then we want to choose rot13 and encrypt. We then enter in the outputfile name which will hold our results. In this case I named it “rot13encryptread” which differentiates it from the other file we created called “rot13encrypt” since it has read which specifies it is read from a file. Then we want to exit the program and navigate to the output file. Here we can see that the message has been encrypted.

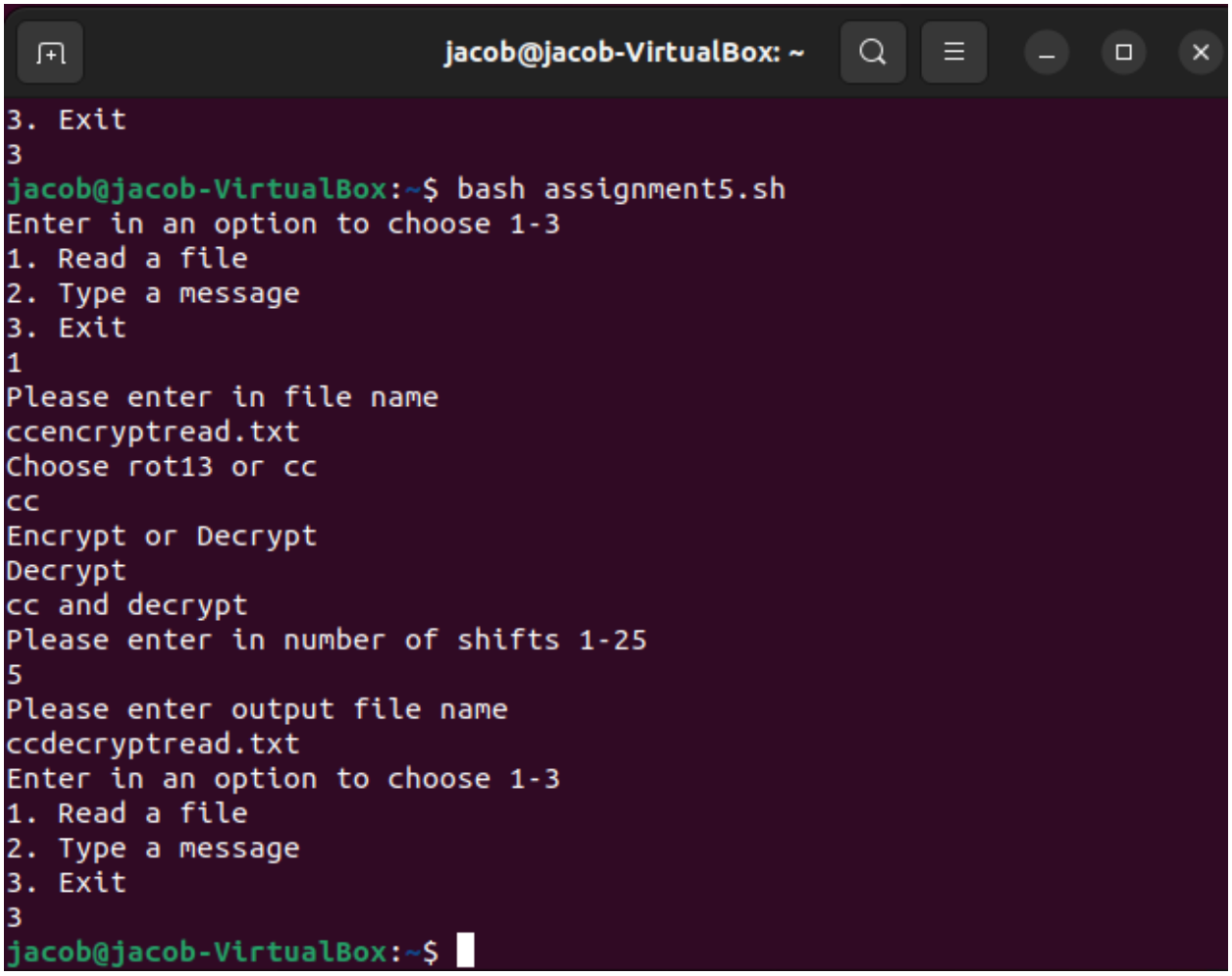
```
jacob@jacob-VirtualBox:~$ bash assignment5.sh
Enter in an option to choose 1-3
1. Read a file
2. Type a message
3. Exit
1
Please enter in file name
rot13encryptread.txt
Choose rot13 or cc
rot13
Encrypt or Decrypt
Decrypt
rot13 and decrypt
Please enter output file name
rot13decryptread.txt
Enter in an option to choose 1-3
1. Read a file
2. Type a message
3. Exit
3
jacob@jacob-VirtualBox:~$
```

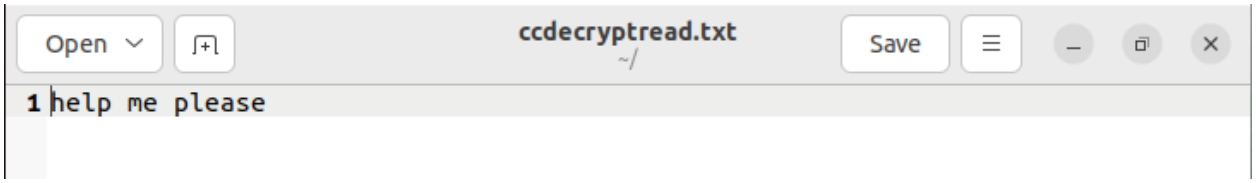
9. Then we want to go back to the terminal and run the script again. We again want to choose option1 and then enter in the name of the encrypted file we just created called “rot13encryptread.txt”. Then we want to choose rot13 and decrypt. After doing so we need to enter in a file name which in this case I chose “rot13decryptread.txt”. Then exit the program and navigate to the output file we just created which should have the decrypted text.

```
jacob@jacob-VirtualBox:~$ bash assignment5.sh
Enter in an option to choose 1-3
1. Read a file
2. Type a message
3. Exit
1
Please enter in file name
readtest.txt
Choose rot13 or cc
cc
Encrypt or Decrypt
Encrypt
cc and encrypt
Please enter in number of shifts 1-25
5
Please enter output file name
ccencryptread.txt
Enter in an option to choose 1-3
1. Read a file
2. Type a message
3. Exit
3
jacob@jacob-VirtualBox:~$
```

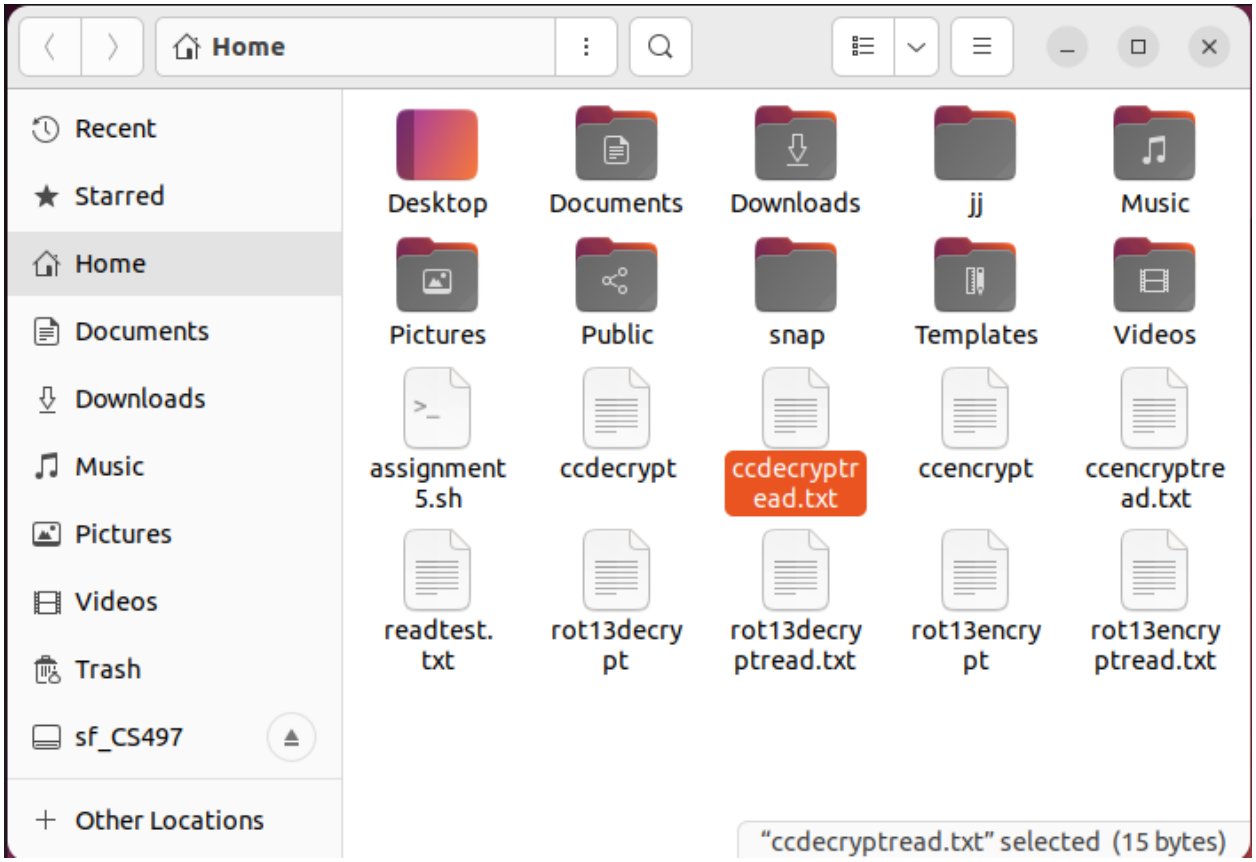


10. Next we want to demonstrate the cc encryption when reading from a file. We want to navigate back to the terminal and begin to run the script again. We then choose option 1, after doing so we enter in the file we created for testing which is called “readtest.txt”. Then we want to choose cc and encrypt. After doing so we need to choose the number of shifts for encryption. We will need to remember this due to decryption later on. Then we enter in the output file name which in this case is “ccencryptread.txt” then we want to exit and navigate to the output file. In the file we should see an encrypted message.





11. Next we then want to navigate back to the terminal again and then we want to choose option 1. After doing so to decrypt we need to enter in the encrypted files name we created within the last step. Then we want to choose cc encryption and the option to decrypt. Then we want to enter in the number of shifts we previously did to encrypt the message. Then we need to choose an output file name which in this case I did “ccdecryptread.txt”. After doing so we exit the program and navigate to the file we just created which should have the decrypted message.



12. When finishing the demonstration we should have a total of 10 files. 1 for the bash shell script and the other 9 text files for the demonstration for the running script.